House Price Prediction Project

1. Problem definition

- Goal: predict the sale price for each house

In [ ]:
```
import numpy as np
import pandas as pd
```

In [ ]:
```
data = pd.read_csv('D:/Programming/Python/Projects/archive (1)/train.csv')
data.head()
```

Out[ ]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandCor
0	1	60	RL	65.0	8450	Pave	NaN	Reg	
1	2	20	RL	80.0	9600	Pave	NaN	Reg	
2	3	60	RL	68.0	11250	Pave	NaN	IR1	
3	4	70	RL	60.0	9550	Pave	NaN	IR1	
4	5	60	RL	84.0	14260	Pave	NaN	IR1	

5 rows × 81 columns

In [ ]:
```
data = pd.read_csv('D:/Programming/Python/Projects/archive (1)/train.csv',index_
data.head()
```

Out[ ]:

	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContou
Id								
1	60	RL	65.0	8450	Pave	NaN	Reg	L
2	20	RL	80.0	9600	Pave	NaN	Reg	L
3	60	RL	68.0	11250	Pave	NaN	IR1	L
4	70	RL	60.0	9550	Pave	NaN	IR1	L
5	60	RL	84.0	14260	Pave	NaN	IR1	L

5 rows × 80 columns

In [ ]:
```
data.columns
```

Out[ ]:
```
Index(['MSSubClass', 'MSZoning', 'LotFrontage', 'LotArea', 'Street', 'Alley',
       'LotShape', 'LandContour', 'Utilities', 'LotConfig', 'LandSlope',
       'Neighborhood', 'Condition1', 'Condition2', 'BldgType', 'HouseStyle',
       'OverallQual', 'OverallCond', 'YearBuilt', 'YearRemodAdd', 'RoofStyle',
       'RoofMatl', 'Exterior1st', 'Exterior2nd', 'MasVnrType', 'MasVnrArea',
       'ExterQual', 'ExterCond', 'Foundation', 'BsmtQual', 'BsmtCond',
       'BsmtExposure', 'BsmtFinType1', 'BsmtFinSF1', 'BsmtFinType2',
       'BsmtFinSF2', 'BsmtUnfSF', 'TotalBsmtSF', 'Heating', 'HeatingQC',
       'CentralAir', 'Electrical', '1stFlrSF', '2ndFlrSF', 'LowQualFinSF',
       'GrLivArea', 'BsmtFullBath', 'BsmtHalfBath', 'FullBath', 'HalfBath',
       'BedroomAbvGr', 'KitchenAbvGr', 'KitchenQual', 'TotRmsAbvGrd',
       'Functional', 'Fireplaces', 'FireplaceQu', 'GarageType', 'GarageYrBlt',
       'GarageFinish', 'GarageCars', 'GarageArea', 'GarageQual', 'GarageCond',
       'PavedDrive', 'WoodDeckSF', 'OpenPorchSF', 'EnclosedPorch', '3SsnPorch',
       'ScreenPorch', 'PoolArea', 'PoolQC', 'Fence', 'MiscFeature', 'MiscVal',
       'MoSold', 'YrSold', 'SaleType', 'SaleCondition', 'SalePrice'],
      dtype='object')
```

2. Feature selections

- Choose features to train ML model
- Need to use "Feature Engineering" to identify features needed

In [ ]:
```
features = ['LotArea','YearBuilt','1stFlrSF','2ndFlrSF','FullBath','BedroomAbvGr
```

3. Spliting datasets

- "data": dataset
- "X": data[features]
- "y": target variable "SalePrice"

In [ ]:
```
X = data[features]
y = data['SalePrice']
data.head()
```

Out[ ]:

Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContou
1	60	RL	65.0	8450	Pave	NaN	Reg	L
2	20	RL	80.0	9600	Pave	NaN	Reg	L
3	60	RL	68.0	11250	Pave	NaN	IR1	L
4	70	RL	60.0	9550	Pave	NaN	IR1	L
5	60	RL	84.0	14260	Pave	NaN	IR1	L

5 rows × 80 columns

In [ ]:
```
X.head()
```

Out[ ]:

Id	LotArea	YearBuilt	1stFlrSF	2ndFlrSF	FullBath	BedroomAbvGr	TotRmsAbvGrd
1	8450	2003	856	854	2	3	8
2	9600	1976	1262	0	2	3	6
3	11250	2001	920	866	2	3	6
4	9550	1915	961	756	1	3	7
5	14260	2000	1145	1053	2	4	9

In [ ]:
```
y.head()
```

Out[ ]:
```
Id
1    208500
2    181500
3    223500
4    140000
5    250000
Name: SalePrice, dtype: int64
```

X, y --> X_train, y_train, X_test, y_test

In [ ]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y, train_size=0.8, test_si
```

In [ ]:
```
X_train.shape
```

Out[ ]:
```
(1168, 7)
```

In [ ]:
```
X_test.shape
```

Out[ ]:
```
(292, 7)
```

4. Training machine learning model

In [ ]:
```
from sklearn.tree import DecisionTreeRegressor

dt_model = DecisionTreeRegressor(random_state=1)
```

- Fit training data into model

In [ ]:
```
dt_model.fit(X_train,y_train)
```

Out[ ]:
```
DecisionTreeRegressor(random_state=1)
```

- Predict

In [ ]:
```
y_pred = dt_model.predict(X_test.head())
```

In [ ]:
```
y_pred
```

Out[ ]:
```
array([335000., 140200., 119000., 207500., 112000.])
```

In [ ]:
```
pd.DataFrame({'y': y_test.head(), 'y_pred': y_pred})
```

Out[ ]:

Id	y	y_pred
530	200624	335000.0
492	133000	140200.0
460	110000	119000.0
280	192000	207500.0
656	88000	112000.0

In [ ]: